IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Mark E. Kriegsman and Benjamin W. Wyckoff	Art Unit	: 2141
		Examiner	: Djenane M. Bayard
Serial No.	: 09/668,110	Conf. No.	: 9580
Filed	: September 22, 2000		
Title	: SERVING DYNAMIC WEB-PAGES		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

The real party in interest is XCELERA, a corporation of Delaware having a place of business at 10 Ashton Drive, Greenwich, CT, as evidenced by an assignment executed April 9, 2001 and recorded at the U.S. Patent Office on July 27, 2004 at Reel 014903, Frame 0179.

(2) Related Appeals and Interferences

There are no related appeals or interference.

(3) Status of Claims

Claims 1-3, 5-17, 19-21, and 23-28 are pending and on appeal. Of these, claims 1, 13, and 19 are independent.

(4) Status of Amendments

All amendments have been entered.

(5) Summary of Claimed Subject Matter

CLAIM 1

A cache server **14** is shown in FIG. 1. According to page 3, lines 24-26, "[a]lthough only a single cache server is shown in FIG. 1, it is understood that the system **10** can include many geographically dispersed cache servers **14**."

A "cache manager (**24**) in communication [a] corresponding cache memory (**26**) for controlling content of said corresponding cache memory" is shown in FIG. 1. A programmable script **36** is shown in FIG. 1 as being accessible to cache manager **24**. The programmable script **36** and its execution by the cache manager (**24**) are described on page 6, lines 4-10. Receiving an update is described on page 6, lines 11-13. The programmability of the script and its role in "detecting the occurrence of a triggering event" are described on page 2, lines 23-28, and page 6, lines 8-12. Causing a particular cache server to request an update of an obsolete portion and receiving that updated portion for storage at the cache server are disclosed at page 4, lines 15-18 of the specification.

CLAIM 3

Claim 3's additional limitation of "interpreting a script containing instructions for defining a rule" is described on page 6, lines 8-10.

CLAIM 5

Claim 5 is an originally filed claim and therefore provides its own support at page 9, lines 17-18.

CLAIM 6

Claim 6's limitation of formulating a database query is disclosed at page 3, lines 17-24.

CLAIM 7

A page controller **34** is shown in both FIGS. 1 and 2 and is described on page 4, lines 10-14, on page 4, lines 15-18, and on page 7, lines 19-21.

CLAIM 8

An origin server **12** in communication with a cache server **14** is shown in FIG. 1 for receiving a request from the cache server **14** and providing an updated portion back to the cache server **14**. The provision of replacement web page objects is discussed between page 3, line 27 and page 4, line 4.

CLAIM 13

A cache server **14** is shown in FIG. 1. According to page 3, lines 24-26, "[a]lthough only a single cache server is shown in FIG. 1, it is understood that the system **10** can include many geographically dispersed cache servers **14**."

A "cache manager (**24**) in communication [a] corresponding cache memory (**26**) for controlling content of said corresponding cache memory" is shown in FIG. 1. A programmable script **36** is shown in FIG. 1 as being accessible to cache manager **24**. The programmable script **36** and its execution by the cache manager (**24**) are described on page 6, lines 4-10. Requesting an update is described on page 6, lines 11-13.

Causing a particular cache server to request an update of an obsolete portion and receiving that updated portion for storage at the cache server are disclosed at page 4, lines 15-18 of the specification.

CLAIM 16

A communication path between an administrator process **20** and a programmable script **36** is shown in FIGS. 1 and 2 and described at page 3, lines 17-20.

CLAIM 17

A page controller **34** is shown in both FIGS. 1 and 2 and is described on page 4, lines 10-14, on page 4, lines 15-18, and on page 7, lines 19-21.

CLAIM 19

A cache server **14** is shown in FIG. 1. According to page 3, lines 24-26, "[a]lthough only a single cache server is shown in FIG. 1, it is understood that the system **10** can include many geographically dispersed cache servers **14**."

A "cache manager (**24**) in communication [a] corresponding cache memory (**26**) for controlling content of said corresponding cache memory" is shown in FIG. 1. A programmable script **36** is shown in FIG. 1 as being accessible to cache manager **24**. The programmable script **36** and its execution by the cache manager (**24**) are described on page 6, lines 4-10. Receiving an update is described on page 6, lines 11-13. The programmability of the script and its role in "detecting the occurrence of a triggering event" are described on page 2, lines 23-28, and page 6, lines 8-12. Causing a particular cache server to request an update of an obsolete portion and receiving that updated portion for storage at the cache server are disclosed at page 4, lines 15-18 of the specification.

CLAIM 21

Claim 3's additional limitation of "interpreting a script containing instructions for defining a rule" is described on page 6, lines 8-10.

CLAIM 23

Claim 23 is an originally filed claim and therefore provides its own support at page 9, lines 17-18.

CLAIM 24

Claim 24's limitation of formulating a database query is disclosed at page 3, lines 17-24.

CLAIM 25

A page controller **34** is shown in both FIGS. 1 and 2 and is described on page 4, lines 10-14, on page 4, lines 15-18, and on page 7, lines 19-21.

CLAIM 26

An origin server **12** in communication with a cache server **14** is shown in FIG. 1 for receiving a request from the cache server **14** and providing an updated portion back to the cache server **14**. The provision of replacement web page objects is discussed between page 3, line 27 and page 4, line 4.

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1, 13, and 19 are rejected as being anticipated under 35 USC 102(*e*) by *Scherr*, US Patent No. 6,799,248.

Claims 3 and 21 are rejected as being anticipated under 35 USC 102(*e*) by *Scherr*, US Patent No. 6,799,248.

Claims 5 and 23 are rejected as being anticipated under 35 USC 102(*e*) by *Scherr*, US Patent No. 6,799,248.

Claims 8 and 26 are rejected as being anticipated under 35 USC 102(*e*) by *Scherr*, US Patent No. 6,799,248.

Claim 16 is rejected as being anticipated under 35 USC 102(*e*) by *Scherr*, US Patent No. 6,799,248.

Claims 6 and 24 stand rejected under 35 USC 103 as being rendered obvious by the combination of *Scherr* and *Kung*, US Patent No. 5,933,837.

Claims 7, 17, and 25 stand rejected as being rendered obvious under 35 USC 103 by the combination of *Scherr* and *Leshem*, US Patent No. 5,870,559.

(7) Argument

TECHNICAL PROBLEM ADDRESSED

At the time the invention was made, it was known that one could more efficiently deliver web pages by storing copies of a web page at many different servers. A user who requested a web page would then be instructed to retrieve that web page from a nearby web server. Since each server would be serving fewer clients, the average wait time for each client would decrease.

Unfortunately, information within a web page can eventually become obsolete. Thus, the foregoing method required some way to ensure that each web server had a current copy of the web page. At the time of the invention, a well-known way to ensure that a web server always had the most current version of the web page was to periodically update the web pages at each web server. For example, an origin server, on which a master copy of the web page was stored, could distribute a new copy of the web page every two hours.

Another known method was to provide each web page with a "stale date." Upon identifying a web page having a stale date behind the current date, a web server would request an updated copy of the web page.

A disadvantage of the foregoing methods for ensuring that web pages remained current was that they were tied strictly to a schedule. Thus, if a web page remained unchanged, it would be updated anyway, simply because a schedule indicated that it was time to do so. This would waste network bandwidth. Conversely, if a web page were to become obsolete, it would not be updated until the next scheduled update.

Thus, at the time the invention was made, a deficiency in the prior art methods of content distribution was that the event that would trigger the update of a web page was tied to a schedule that did not necessarily reflect whether or not the web page actually needed to be updated at all.

THE PRIMARY REFERENCE

Scherr[1] discloses a cache management system that chooses among several cache management algorithms.[2] Among these algorithms is one called the "time currency method." *Scherr* describes the "time currency method" as a cache management method in which pages are periodically refreshed. Specifically, *Scherr* states that

> "[a] time currency method of cache management can be configured to refresh certain pages with one frequency, say every 15 minutes, during trading hours for a given exchange, and with another frequency, say until start of trading the next trading day, once the exchange has closed."[3]

What appears to be a script for executing this "time currency method" is shown at *Scherr*'s FIG. 12:

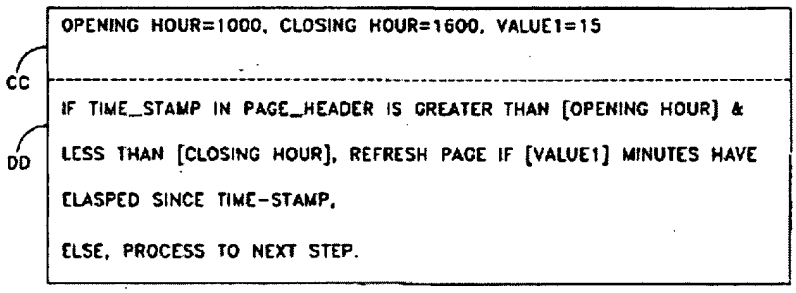

FIG. 12

Thus, based on both FIG. 12 and the above description from *Scherr*, the time-currency method appears to be a way to refresh a web page at periodic intervals. These periodic intervals do not depend on whether or not the web page is in fact obsolete. They simply depend on the passage of time.

SECTION 102 REJECTION OF CLAIMS 1, 13, AND 19

As best understood, the Examiner regards claim 1's "triggering event" as corresponding to the passage of a time interval, such as "[VALUE 1]" in *Scherr*'s FIG. 12 (see above). For

---

[1] *Scherr*, U.S. Patent No. 6,799,248.
[2] see *Scherr*, claim 1.
[3] *Scherr*, col. 6, lines 18-23.

example, in the Examiner's view, when 15 minutes go by during trading hours (as discussed in *Scherr*, col. 6, lines 18-24), a "triggering event" has occurred.

However, claim 1 does not recite just any triggering event. It recites a particular species of triggering event. In particular, claim 1 recites only those triggering events that are "indicative of the existence of an obsolete portion of a web-page stored in [a] cache server."

The lapse of, for example, fifteen minutes has nothing to do with whether or not a portion of a web page is obsolete. It may be that a web page will remain unchanged for fifteen minutes. Or, the web page may become obsolete several times within fifteen minutes. Thus, the triggering event:

"a pre-selected time interval has lapsed"

is *not* a triggering event "indicative of the existence of an obsolete portion" of a web page as required by claim 1.

In response to Applicant's position, the Examiner states that

> "*Scherr* clearly teaches wherein the cache management system could be configured to pre-fetch web pages from the request site each time an internal user logs on and those pages are not already in cache storage."[4]

In support of this position, the Examiner appears to rely on the following text from *Scherr*:

> "For example, if most of the internal users are likely to request pages from the same website, when they first log on, cache management system **10** at local site **06** could be configured to pre-fetch web pages from the requested site each time an internal user logs on and those pages are not already in cache storage."[5]

The cited portion of *Scherr*[6] merely states that if one knows in advance that a particular user is likely to request a particular web page, it is a good idea to have that web page ready in cache for the user. In the foregoing passage, *Scherr* discloses a new triggering event. Instead of the triggering event:

"a pre-selected time interval has lapsed,"

the new triggering event is

---

[4] *Final Office Action*, page 2.
[5] *Scherr*, col. 6, lines 28-34.
[6] *Scherr*, col. 6, lines 27-32.

"a user has logged on."

But a user's act of logging on is completely unrelated to whether or not a particular web page has an obsolete portion. Users routinely log on and off a system whenever they wish. Therefore, the triggering event "a user has logged on" cannot possibly be regarded as "indicative of the existence of an obsolete portion" of any web page.

Accordingly, neither the passage of a pre-selected time nor the log-on of a user are events that correspond to claim 1's recitation of a triggering event "indicative of the existence of an obsolete portion" of a web page.

Claims 13 and 19 include limitations similar to claim 1. Accordingly, those claims are patentable for at least the same reasons as claim 1.

SECTION 102 REJECTION OF CLAIM 3 AND 21

Claim 3 further limits claim 1's step of "implementing programmable rules...defining a triggering event...indicative of the existence of an obsolete portion of said web page" to require "interpreting a script containing instructions for defining a rule."

The Examiner states that claim 3's additional limitation is disclosed in the following passage from *Scherr*:

> "Depending on the configuration(s) selected, the system may manage data or subsets of data in a storage cache on the basis of time-currency, page usage frequency, charging considerations, pre-fetching algorithms, data-usage patterns, store-through methods for updated pages, least recently used method, B-tree algorithms, or indexing techniques including named element ordering, among others. A preferred embodiment may embed the configurable cache management in the storage media, either as firmware in a storage controller or as software executing in a central processing unit (CPU) in a storage controller. In a preferred embodiment the system may be scaled in size and offer security for protected data."[7]

The cited passage inventories the various cache-management methods available. This passage appears unrelated to claim 3's limitation of "interpreting a script."

[7] *Scherr*, col. 3, lines 54-65.

Scherr also describes a cache management monitor that responds to a message transmitted to it by a script, such as a usage-pattern analyzer.[8] However, *Scherr*'s usage-pattern analyzer has nothing to do with "defining a triggering event" the occurrence of which is "indicative of the existence of an obsolete portion of the web page."

As best understood, *Scherr*'s usage-pattern analyzer instructs the cache manager to pre-fetch certain pages because, based on an analysis of usage patterns, those pages are likely to be requested soon. Whether or not those pages are obsolete is irrelevant to the usage-pattern analyzer. Therefore, to the extent a script associated with the *Scherr* usage-pattern analyzer defines a triggering event, that triggering event has nothing to do with web page obsolescence.

Scherr refers to additional scripts at col. 7, lines 5-9 in connection with FIGS. 11 and 12. But neither script is related to claim 3's additional limitation.

FIG. 11 shows a script that pre-fetches an existing web page when two variables match. FIG. 12 shows a script that refreshes a web page after the passage of some time interval. Neither one has anything to do with defining a triggering event indicative of an obsolete portion of a web page.

Claim 21 includes limitations similar to claim 3 and is patentable for at least the same reasons.

SECTION 102(*e*) REJECTION OF CLAIMS 5 and 23

Claim 5 recites the additional limitation that detecting a triggering event include "detecting the receipt of an updated portion of said web page."

The Examiner states that this limitation is disclosed by *Scherr* at column 11, lines 53-56.[9] This passage, and its surrounding text, read as follows:

> "Still another form of indexing or pre-fetching that could be used in an alternative preferred embodiment of the present invention is the technique known as mirroring. If users at a local site are constantly accessing a large website located outside the country, the cache management methods of the present invention might create a local mirror of that site in storage units **14**, and use the protocols provided by the source for updating the mirror image. These normally include an initial transfer of all data using a file transfer protocol (FTP)--like protocol, and then regularly scheduled updates that cause any changes made at the source site to be transferred to the mirror. Where the local site has a large amount of storage available for storage units **14**, the present

[8] *Scherr*, col. 7, lines 16-19. ("As another example, the monitor used for cache management could response to messages transmitted to it by a program or script running at the same site, such as a usage pattern analyzer.")
[9] *Final Action*, page 5.

> invention could include several mirrors in the cache as well as other indexes. Additionally, service providers could offer supplying the mirror files as one of their services. In this approach, updates would be sent to a local site by the service provider as they occur and without being solicited by a file transfer request from the local cache management system **10**."[10]

The above passage merely describes a service for pushing an updated file from a source site to a mirror site according to "regularly scheduled updates." There is no suggestion in the foregoing passage that the mirror site detects any triggering event that indicates an obsolete portion of a web page.

Accordingly, the foregoing passage fails to describe detecting a triggering event in a way that includes detecting an updated portion of a web page as recited in claim 5.

The Examiner further suggests that "detecting the receipt of an updated portion of said web page" is disclosed in the following passage from *Scherr*:[11]

> "Depending on the configuration(s) selected, the system may manage data or subsets of data in a storage cache on the basis of time-currency, page usage frequency, charging considerations, pre-fetching algorithms, data-usage patterns, store-through methods for updated pages, least recently used method, B-tree algorithms, or indexing techniques including named element ordering, among others. A preferred embodiment may embed the configurable cache management in the storage media, either as firmware in a storage controller or as software executing in a central processing unit (CPU) in a storage controller. In a preferred embodiment the system may be scaled in size and offer security for protected data."[12]

But the above passage merely: (1) states that *Scherr*'s system can be configured to use many different cache management algorithms; and (2) inventories the available algorithms, none of which appear to be pertinent to the claim limitation. The fact that a system can be configured to use many cache management algorithms does not amount to a disclosure of a particular algorithm in which detecting a "triggering event" includes "detecting the receipt of an updated portion of said web page."

Claim 23 includes limitations similar to claim 5 and is patentable for at least the same reasons.

---

[10] *Scherr*, col. 11, lines 38-57.
[11] *Final Action*, page 3.
[12] *Scherr*, col. 3, lines 54-65.

SECTION 102(*e*) OF CLAIMS 8 AND 26

Claim 8 recites the additional limitation of causing a cache server to request an update from an origin server, and receiving an updated portion from that origin server.

In rejecting claim 8, the Examiner asserts that both of the above limitations are disclosed by *Scherr* at col. 11, lines 52-56.[13] This passage, and its surrounding text read as follows:

> "Still another form of indexing or pre-fetching that could be used in an alternative preferred embodiment of the present invention is the technique known as mirroring. If users at a local site are constantly accessing a large website located outside the country, the cache management methods of the present invention might create a local mirror of that site in storage units **14**, and use the protocols provided by the source for updating the mirror image. These normally include an initial transfer of all data using a file transfer protocol (FTP)--like protocol, and then *regularly scheduled updates that cause any changes made at the source site to be transferred to the mirror.* Where the local site has a large amount of storage available for storage units **14**, the present invention could include several mirrors in the cache as well as other indexes. Additionally, service providers could offer supplying the mirror files as one of their services. In this approach, updates would be sent to a local site by the service provider as they occur and without being solicited by a file transfer request from the local cache management system **10**."[14]

As best understood, the Examiner regards *Scherr*'s source site as being the claimed "origin server," since that is where the "update" of claim 8 would come from. In that case, *Scherr*'s "mirror site" would have to be the claim's "cache-server." But for this mapping to truly read on claim 8. *Scherr*'s "mirror site" would have to actually request content from *Scherr*'s source site.

Scherr's mirror site does *not* request content from the source site at all. In fact, precisely the opposite happens. *Scherr*'s source site pushes content to *Scherr*'s mirror site in "regularly scheduled updates." Thus, *Scherr*'s mirror site is passive. It does not request or pull content from *Scherr*'s source site. As a result, the foregoing fails to teach the additional limitation of claim 8.

---

[13] *Final Office Action*, page 3.
[14] *Scherr*, col. 11, lines 38-57. [emphasis supplied]

The Examiner further suggests that *Scherr* discloses the limitation of requesting an update portion from an origin server and receiving the updated portion from that origin server is disclosed in the following passage from *Scherr*:[15]

> "If, in FIG. 2a, at decision block **30** it is determined that the data is not already in the cache (here, in storage units **14**), a request will be made to fetch the data from the network at step **34**."[16]

The foregoing passage merely describes the idea that if requested data is not already in cache it should be retrieved from the network. This has nothing to do with whether or not the requested data is obsolete. The algorithm shown in FIG. 2A does not concern itself with whether data is obsolete. The algorithm in FIG. 2A only concerns itself with whether or not the requested data is even available in the first place.

SECTION 102 (e) REJECTION OF CLAIM 16

Claim 16 recites the additional limitation that the claimed web-serving system include "a communication path between said administrative process and said programmable script." This additional limitation is allegedly described within the following text[17]

> "Still in FIG. la, the administrator of backbone link **04** might prefer to configure its cache management system **10** to use page usage or data usage patterns for providing the best overall response times."[18]

The foregoing passage describes an administrator, either a human or an expert system, whose job is to choose a particular cache management algorithm for achieving the best response times. The *Scherr* administrator is not in communication with any programmable script. It simply selects one among several cache management algorithms.

SECTION 103 REJECTION OF CLAIMS 6 AND 24

Claim 6 recites the additional limitation that requesting an update include "formulating a database query."

---

[15] *Final Office Action*, page 5.
[16] *Scherr*, col. 8, lines 44-47.
[17] *Final Action*, page 6.
[18] *Scherr*, col. 5, lines 62-65.

The Examiner has already conceded that *Scherr* fails to disclose the additional limitation of formulating a database query. Nevertheless, the Examiner regards it as obvious for one of ordinary skill in the art to have modified *Scherr*'s teachings as directed by *Kung*.

Kung describes a system in which a primary database and numerous subscribing databases are connected by a network. In *Kung*'s system, when a change is made to the primary database, that change must also be made at each of the subscribing databases.

The Examiner presumably regards *Kung*'s "primary database" as being like an "origin server" and *Kung*'s "subscribing databases" as being like Applicant's "cache servers." The update of the primary database would thus be a "triggering event" that triggers the update of all the subscribing databases.

The Examiner regards claim 6's additional limitation as being disclosed somewhere in the following passage:[19]

> "These and other objects of the invention are accomplished in accordance with the principles of the invention by providing a database network having a primary database and a plurality of heterogeneous subscribing databases for replicating data updates of the primary database in the heterogeneous subscribing databases. The database network includes a primary database engine connected to the primary database for capturing the data updates in the primary database. A query manager connected to the database engine for generating queries translates queries based on a specified format for each of the heterogeneous subscribing databases. A data distributor connected to the query manager distributes the translated queries to the heterogeneous subscribing databases.
>
> "The database network may further include a subscription controller for verifying that at least one of said subscribing databases needs the updates made in the primary database so that the query manager generates queries only for the subscribing databases that need the updates.
>
> "The database network may also include a synchronization manager for controlling sequences of the queue distribution where each queue is categorized as immediate or deferred queue based on query's time sensitivity. The queue may be categorized as a retry queue if the previous transmission has failed."[20]

According to the foregoing passage, database queries are generated by a "query manager" at the primary database and distributed to subscribing databases. This is further apparent from

---

[19] *Final Action*, page 7.
[20] *Kung*, col. 2, lines 39-64.

FIG. 2, which shows the query manager **202** as being part of the data consistency server **114**, which is part of the primary system **102** in FIG. 1.[21]

It is apparent therefore that in *Kung*, the act of "formulating a database query" is carried out at the *primary* system **102**. But according to claim 6, the act of "formulating a database query" must be part of "requesting an update." In *Kung*, the primary system **102** does not *request* updates, it *provides* updates. Therefore, when the primary system **102** formulates a database query, it is *not* requesting an update, as required by claim 6.

As best understood, in *Kung*, the primary system **102** ultimately provides the database query to the subscribing systems **104**. Thus, although a subscribing system may ultimately *use* the database query in connection with requesting an update, the subscribing system does not actually *generate* the database query as required by claim 6.

Motivation to combine *Scherr* and *Kung* is flawed

As motivation to combine *Scherr* and *Kung*, the Examiner suggests that one of ordinary skill in the art would have been motivated to modify *Scherr* as disclosed by *Kung* to "accurately and promptly synchronize heterogeneous databases."

But the cache servers of *Scherr* have no databases to update. One of ordinary skill in the art would have had no reason to modify *Scherr* to update a database because he would have seen immediately that *Scherr* had no databases to update.

The Examiner's proposed motivation to combine *Scherr* and *Kung* is essentially a restatement of the technical problem *Kung* faced, followed immediately thereafter by *Kung*'s solution to the same problem. It has no relationship to *Scherr*'s cache management system. Stripped to its logical essentials, the Examiner's position is that "it would have been obvious to modify *Scherr* as disclosed by *Kung* in order to do what *Kung* already does all by itself."

---

[21] *Kung*, col. 3, lines 38-45 "FIG. 2 shows integrated data consistency server **114** (FIG. 1) connected to primary database engine **110** (FIG. 1) and communication network **106** (FIG. 1) in more details. Integrated data consistency server **114** includes subscription controller ("SC") **200**, query manager ("QM") **204** , update queues ("Un") **206**, queue controllers ("Qc") **208**, concurrence controller ("CC") **210**, data distributor ("DD") **212** and response handler ("RH") **214**."

Such a statement could be made about any pair of references whatsoever. Accordingly, it can hardly amount to "a convincing line of reasoning as to why the artisan would have found the claimed invention obvious in light of the references."[22]

Claim 24 includes limitations similar to claim 6 and is patentable for at least the same reasons.

SECTION 103 REJECTION OF CLAIMS 7, 17, AND 25

Claim 7 recites the additional limitation of "providing an assembly script containing instructions for assembling constituent portions of a web page and said updated portion into an updated web page."

The Examiner has already conceded *Scherr*'s failure to disclose any such assembly scripts. Nevertheless, the Examiner asserts that somewhere in the following passage, *Lesham* teaches providing such an assembly script:

> "In accordance with yet another aspect of the invention, the Web site analysis program includes software routines and associated user interface controls for automatically scanning and mapping dynamically-generated Web pages, such as Web pages generated "on-the-fly" in response to user-specified database queries. This feature generally involves the two-step process of capturing and recording a dataset manually entered by the user into an embedded form of a Web page (such as a page of a previously-mapped Web site), and then automatically resubmitting the dataset (within the form) when the Web site is later re-scanned. As will be appreciated, this feature of the invention can also be applied to conventional Internet search engines.
>
> To effectuate the capture of one or more datasets in the preferred implementation, the user initiates a capture session from the user interface; this causes a standard Web browser to be launched and temporarily configured to use the Web site analysis program as an HTTP-level proxy to communicate with Web sites. Thereafter, until the capture session is terminated by the user, any pages retrieved with the browser, and any forms (datasets) submitted from the browser, are automatically recorded by the Web site analysis program into the site map. When the site map is subsequently updated (using an "automatic update" option of the user interface), the scanning routines automatically re-enter the captured datasets into the corresponding forms and recreate the form submissions. The dynamically-generated Web pages returned in response to these automatic form submissions are then added to the updated site map as respective nodes. A related aspect of the invention involves the associated method of locally capturing the output of the Web browser to generate a sequence that can subsequently be used to automatically evaluate a Web site."[23]

---

[22] *Ex Parte Clapp*, 227 USPQ 972, 973 (BPAI 1985).
[23] *Lesham*, col. 3, lines 30-63.

Lesham teaches a software system that creates a "site map" for a web site. A typical site map, shown in FIG. 1, graphically displays hypertext links associated with that web site.

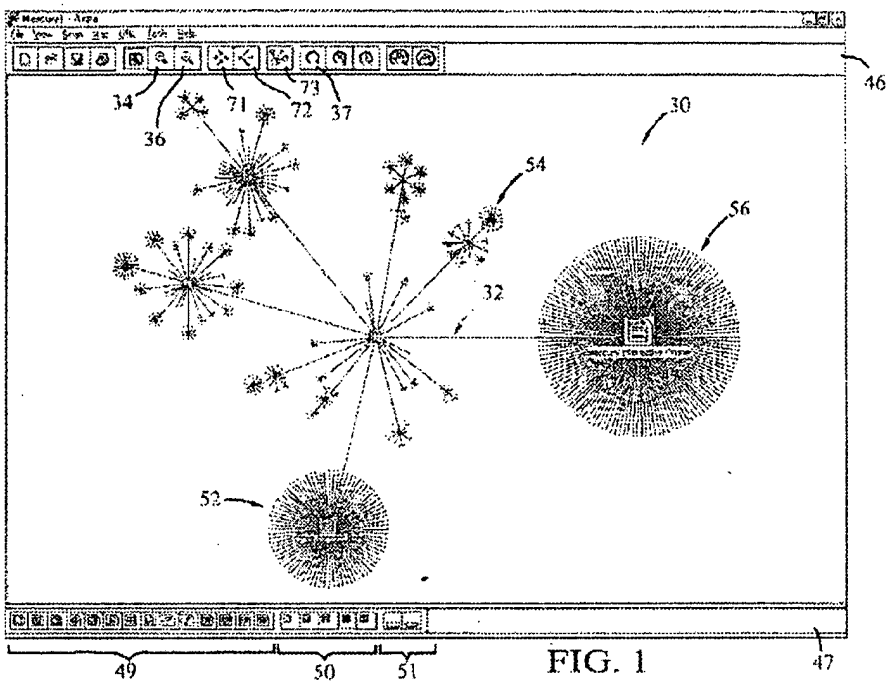

FIG. 1

In some cases, the web site includes a dynamically generated web page. The foregoing text describes how *Lesham*'s system creates a site map for a dynamically generated web page.

It is unclear how the foregoing passage teaches "providing an assembly script containing instructions for assembling constituent portions of a web page and [an] updated portion [of a web page] into an update web page."

The cited passage describes capturing data previously entered in a form, and re-entering that captured data into a dynamically-generated web page. This appears to be carried out when updating the graphical display of a web site's link structure, or "site map," such as that shown in FIG. 1. Applicant speculates that the Examiner has been led to regard this concept as having something to do with claim 7's "assembly script."

However, what *Lesham* discloses amounts to little more than automatically filling out a form. According to claim 1 the "updated portion" is something a cache server would request in response to a "triggering event" that indicates the existence of an obsolete portion of the web page. The data one enters into a form provided by a web page is therefore not an "updated portion" as recited in claim 7.

The data one enters into a form provided by the web page is not part of the web page. However, even if it were, it is not an obsolete portion requested in response to any triggering event as required by claim 1, from which claim 7 depends.

Motivation to combine *Lesham* and *Scherr* is flawed

Lesham describes a tool that a web-page designer might use to visually inspect the hyperlinks associated with a web site.

Scherr describes a cache management system for enabling a server to decide which web pages should be kept in cache for easy access, and which should not.

The test for obviousness is what the combined teachings of the references would have suggested to one of ordinary skill in the art.[24] The Examiner can satisfy this burden by showing some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.[25]

The Examiner asserts that one of ordinary skill in the art would have found it obvious to modify *Scherr* as disclosed by *Lesham* "in order to facilitate the management and analysis of WWW sites and other types of database systems which utilize hyperlinks to facilitate user navigation."[26]

The Examiner's motivation to combine the references is taken essentially verbatim from *Lesham*'s statement on the field of his invention:

---

[24] See *In re Kahn*, 441 F.3d 977, 987-88 (Fed. Cir. 2006), *In re Young*, 927 F.2d 588, 591 (Fed. Cir. 1991), and *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).
[25] *KSR Int'l v. Teleflex Inc.*, 127 S. Ct. 1727, 1741 (2007) (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).
[26] *Final Action*, page 8.

FIELD OF THE INVENTION

The present invention relates to database management and analysis tools. More particularly, the present invention relates to software tools for facilitating the management and analysis of World Wide Web sites and other types of database systems which utilize hyperlinks to facilitate user navigation.

The proposed motivation to modify *Scherr* does no more than suggest what problems *Lesham* set out to solve. There is no apparent connection between the foregoing passage and any technical problem faced by *Scherr*'s cache management system.

In effect, the Examiner's logic, stripped to its essentials is: "it would have been obvious to modify *Scherr* as taught by *Lesham* in order to do what *Lesham* already does all by itself."

The foregoing logic creates a pastiche of two references with no apparent nexus between them. Such reasoning hardly rises to the level of "a convincing line of reasoning as to why the artisan would have found the invention obvious in light of the references."[27]

In fact, there would be no reason for one of ordinary skill in the art to even consider *Lesham* when addressing the technical problem posed by *Scherr*.

Scherr addresses the problem of deciding what files to cache for easy access and what files to leave out of cache. It makes no difference to *Scherr* whether these files have hyperlinks. In fact, it makes not difference to *Scherr* whether those files are even web pages. *Scherr* is concerned with caching files in such a way that those files reach the user promptly.

Accordingly, the proposed motivation for combining references is flawed, and fails to raise a prima facie case of obviousness.

Claims 17 and 25 include limitations similar to claim 7 and are patentable for at least the same reasons.

(8) Conclusion

[27] *Ex Parte Clapp*, 227 USPQ 972, 973 (BPAI 1985).

Please apply the brief fee of $255, along with any other charges or credits to Deposit Account No. 06-1050, referencing Attorney Docket No. 11125-014001.

Respectfully submitted,

Date: July 15, 2008

/Faustino A. Lichauco/
Faustino A. Lichauco
Reg. No. 41,942

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (877) 769-7945

21948993 (2).doc

Appendix of Claims

1. A method for enabling the generation of an updated web-page for storage in one of a plurality of cache servers, said method comprising:

implementing programmable rules executing on each of the plurality of cache servers, each programmable rule defining a triggering event associated with its corresponding cache server, the occurrence of the triggering event being indicative of the existence of an obsolete portion of said web-page stored in said corresponding cache server;

detecting an occurrence of a triggering event at a particular cache server selected from the plurality of cache servers;

in response to the occurrence of said triggering event, causing said particular cache server to request an update of said obsolete portion; and

receiving an updated portion of said web-page for storage at said particular cache server.

2. The method of claim **1**, further comprising

generating a web-page incorporating said updated portion therein; and

serving said web-page to a user.

3. The method of claim **1**, wherein implementing said programmable rules comprises interpreting a script containing instructions for defining a rule.

4. (Cancelled)

5. The method of claim **1**, wherein detecting said triggering event comprises detecting the receipt of an updated portion of said web-page.

6. The method of claim **1**, wherein requesting an update comprises formulating a database query to be carried out by a database engine.

7. The method of claim **1**, wherein said method further comprises providing an assembly script containing instructions for assembling constituent portions of a web-page and said updated portion into an updated web-page.

8. The method of claim **1**, wherein

causing said particular cache-server to request an update comprises establishing communication with an origin server and causing said particular cache server to request said update therefrom, and

receiving an updated portion comprises receiving said updated portion from said origin server.

9. The method of claim **8**, further comprising providing a cache memory element separate from said origin server.

10. The method of claim **8**, further comprising providing a cache memory element at said origin server.

11. The method of claim **1**, further comprising collecting access-data indicative of how frequently said web-page is requested.

12. The method of claim **11**, further comprising managing the content of caches in said cache servers in response to said access-data.

13. A web-serving system comprising:

a plurality of cache servers each having a corresponding cache memory; and

a cache manager in communication with said corresponding cache memory for controlling content of said corresponding cache memory, said cache manager being configured to execute a programmable script, said script being configured for

detecting the occurrence of a triggering event, and in response to detection of said triggering event, causing said cache manager to request an update of said content of said cache memory.

14. The web-serving system of claim **13**, further comprising a usage-monitor for collecting access-data indicative of the frequency with which a selected web-page is requested.

15. The web-serving system of claim **14**, wherein said usage-monitor provides said access data to said programmable script, and said programmable script alters content of said corresponding cache memory in response to said access-data.

16. The web-serving system of claim **13**, further comprising a communication path between said programmable script and an administrator process, said communication path enabling said programmable script to receive instructions from said administrator process.

17. The web-serving system of claim **13**, further comprising a page assembler executing on said cache server, the page assembler containing instructions for assembling constituent portions of a web-page into a web-page.

18. (Cancelled)

19. A computer-readable medium having encoded thereon software for updating web-pages stored in caches, each cache being associated with a corresponding cache server from a plurality of cache serves, said software comprising instructions for:

implementing programmable rules executing on each of the plurality of cache servers, each programmable rule defining a triggering event associated with its corresponding cache server, the occurrence of the triggering event being indicative

of the existence of an obsolete portion of said web-page stored in said corresponding cache server;

detecting an occurrence of a triggering event at a particular cache server selected from the plurality of cache servers;

in response to the occurrence of said triggering event, causing said particular cache server to request an update of said obsolete portion; and

receiving an updated portion of said web-page for storage at said particular cache server.

20. The computer-readable medium of claim **19**, wherein said software further comprises instructions for:

generating a web-page incorporating said updated portion therein; and

serving said web-page to a user.

21. The computer-readable medium of claim **19**, said instructions for implementing said first programmable rule further comprise instructions for interpreting a script containing instructions for defining a first programmable rule.

22. (Cancelled)

23. The computer-readable medium of claim **19**, wherein said instructions for detecting said triggering event comprise instructions detecting the receipt of an updated portion of said web-page.

24. The computer-readable medium of claim **19**, wherein said instructions for requesting an updated portion of said web page comprise instructions for formulating a database query to be carried out by a database engine.

25. The computer-readable medium of claim **19**, said computer-readable medium further

comprises instructions for assembling constituent portions of a web-page and said updated portion into an updated web-page.

26. The computer-readable medium of claim **19**, wherein

said instructions for causing said particular cache server to request an update comprise instructions for establishing communication with an origin server and requesting said update therefrom, and

said instructions for causing said particular cache server to receive an updated portion comprise instructions for receiving said updated portion from said origin server.

27. The computer-readable medium of claim **19**, wherein said software further comprises instructions for collecting access-data indicative of how frequently said web-page is requested.

28. The computer-readable medium of claim **27**, wherein said software further comprises instructions for managing the content of said caches in response to said access-data.

Evidence Appendix

None.

Related Proceedings Appendix

None.